

Optimasi Rekomendasi Kafe Terdekat Menggunakan Teori Graf Berdasarkan Jarak dan Rating di Daerah Jatinangor

Mahesa Fadhillah Andre - 13523140

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

mahesa0208@gmail.com, 13523140@std.stei.itb.ac.id

Abstrak—Pemilihan kafe pada Jatinangor merupakan sebuah tantangan tersendiri. Penelitian ini mengembangkan sistem rekomendasi kafe terdekat menggunakan teori graf dalam matematika diskrit. Sistem rekomendasi dioptimalisasi menggunakan pembobotan yang memadukan rating, jarak, dan jumlah rating kafe. Sistem rekomendasi dibuat berdasarkan upagraf dengan pembobotan minimal untuk menggambarkan lintasan terdekat dari titik awal menuju setiap kafe secara satu per satu.

Kata kunci—optimasi, rekomendasi, Jatinangor, bobot

I. PENDAHULUAN

Jatinangor adalah sebuah kecamatan di Kabupaten Sumedang, Provinsi Jawa Barat, Indonesia. Kecamatan Jatinangor terkenal sebagai daerah yang penuh dengan mahasiswa. Sebab, terdapat empat perguruan tinggi negeri yang letaknya di Jatinangor, yaitu Institut Teknologi Bandung (ITB), Universitas Padjadjaran (UNPAD), Institut Pemerintahan Dalam Negeri (IPDN), dan Institut Manajemen Koperasi Indonesia (Ikopin).

Total penduduk Jatinangor yang merupakan mahasiswa di Jatinangor berada pada kisaran 50.000, sedangkan total penduduk Jatinangor berdasarkan data Badan Pusat Statistik Kabupaten Sumedang pada tahun 2022 mencapai angka 98.000. Dengan mayoritas penduduk Jatinangor adalah mahasiswa, gaya hidup dan ekonomi pada daerah tersebut menyesuaikan dengan kebutuhan dan preferensi mayoritas penduduknya.

Dari Agustus 2023 hingga Desember 2024, telah dibuka sejumlah kafe dan coffee shop baru seperti Fore, Bagi Kopi, dan Kopi Toleransi yang merupakan coffee shop dan kafe ternama. Pembukaan coffee shop dan kafe tersebut disebabkan oleh peluang bisnis yang muncul dari perilaku mahasiswa yang suka pergi ke coffee shop dan kafe.

Mahasiswa pada umumnya suka menghabiskan waktunya di kafe karena kafe menyediakan suasana yang nyaman untuk berbagai kegiatan akademik dan sosial. Kafe sering sekali menjadi tempat yang nyaman untuk belajar dan bersosialisasi. Mayoritas kafe di Jatinangor menyediakan fasilitas seperti stop kontak untuk mengecas alat elektronik, ruangan ber-A/C dan/atau area merokok, jam operasional yang melampaui tengah malam (walaupun tidak semua), serta menu dan suasana yang menarik.

Dengan banyaknya kafe di Jatinangor, dibutuhkan sistem rekomendasi kafe yang optimal untuk memudahkan pemilihan kafe yang paling sesuai dengan preferensi mahasiswa/penduduk. Pada makalah ini, rekomendasi kafe didasarkan pada rating yang tersedia untuk setiap kafe pada aplikasi google maps serta jarak dari lokasi pengguna terhadap setiap kafe. Optimalisasi ini memanfaatkan teori graf yang diberi pada mata kuliah matematika diskrit IF1220.

II. LANDASAN TEORI

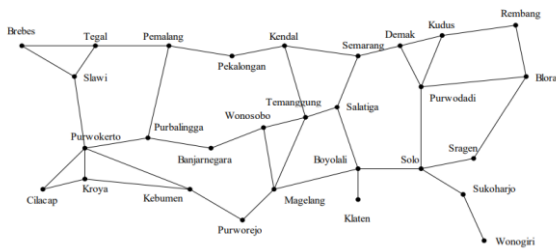
A. Graf

Graf dalam teori graf matematika diskrit adalah sebuah representasi terstruktur yang menggambarkan hubungan sejumlah objek diskrit dengan objek-objek diskrit lainnya. Graf terdiri dari komponen simpul (*node*) dan sisi (*edges*).

Simpul/*vertices* dalam graf direpresentasikan sebagai sebuah titik. Simpul dalam graf berguna untuk merepresentasikan sebuah objek diskrit yang memiliki hubungan dengan objek diskrit lainnya. Contohnya, pada sebuah graf peta Jawa Timur, simpul-simpul merepresentasikan sebuah kota (digambarkan sebagai sebuah titik dengan simpul nama salah satu kota Jawa Timur). Contoh lain, pada aplikasi sosial Instagram, graf dapat digunakan untuk merepresentasikan jaringan followers pada sebuah akun.

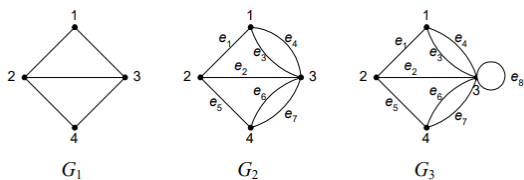
Sisi/*edges* dalam graf digambarkan sebagai sebuah garis yang menghubungkan satu simpul dengan simpul yang lain. Sebuah sisi berguna untuk merepresentasikan sebuah hubungan dari satu simpul dengan sebuah simpul lain. Oleh karena itu, apabila dua buah simpul tidak terhubung oleh sebuah sisi, maka dua buah simpul tersebut tidak terhubung (secara langsung).

Pada teori graf, graf dapat ditulis sebagai $G = (V, E)$ dengan V adalah simpul dan E adalah sisi).



Gambar 2.1 Graf yang menggambarkan jaringan jalan raya yang menghubungkan kota-kota di Provinsi Jawa Tengah.

Sumber:
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>



Gambar 2.2 Beberapa contoh graf.

Sumber:
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

B. Jenis Graf

Berdasarkan terdapat atau tidaknya gelang ganda (sisi yang menghubungkan sebuah simpul dengan dirinya sendiri) dan sisi ganda, graf dibagi menjadi dua jenis:

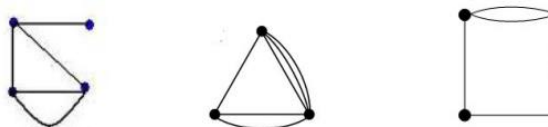
1) Graf sederhana

Graf sederhana adalah graf yang tidak memiliki simpul yang memiliki sisi ganda dengan simpul lain dan juga tidak memiliki simpul yang terhubung kepada dirinya sendiri. Pada gambar 2.2, graf G1 merupakan graf sederhana.

2) Graf tak-sederhana

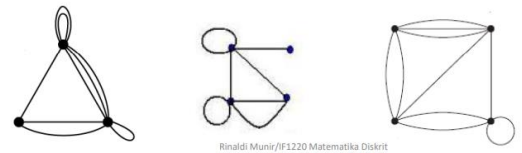
Graf tak-sederhana adalah graf yang memiliki simpul bersisi ganda dan/atau memiliki gelang ganda. Pada gambar 2.2, dapat dilihat Graf G2 dan G3 yang merupakan graf tak-sederhana

Graf tak-sederhana dibagi menjadi dua jenis lagi berdasarkan ada atau tidaknya sisi ganda atau sisi bergelang ganda.



Gambar 2.3 Graf ganda (multi-graph), graf yang memiliki sisi ganda.

Sumber:
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>



Gambar 2.4 Graf semu (pseudo-graph), graf yang memiliki gelang ganda.

Sumber:
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

Berdasarkan terdapat atau tidaknya orientasi arah apada sisi, graf dibagi menjadi dua jenis:

1) Graf tak-berarah (undirected graph)

Graf tak-berarah adalah graf yang terdiri dari sisi-sisi yang tidak memiliki orientasi arah.

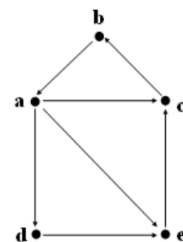
2) Graf berarah (directed graph)

Graf berarah adalah graf yang terdiri dari sisi-sisi yang memiliki orientasi arah.

C. Terminologi di dalam graf

1) Ketetanggaan (*Adjacency*)

Apabila sebuah simpul terhubung langsung dengan simpul lain, maka simpul tersebut bertetangga dengan simpul lain. Misal terdapat sebuah simpul A dan simpul A terhubung dengan simpul B dan C. Maka, simpul A bertetangga dengan simpul B dan C.



Gambar 2.5 Contoh ketetanggaan.

Sumber:
<https://mti.binus.ac.id/2018/03/05/teori-graph-sejarah-dan-manfaatnya/>

2) Bersisian (*Incidency*)

Apabila sebuah sisi E menghubungkan dua buah simpul sembarang a dan b, maka sisi E bersisian dengan simpul a dan E bersisian dengan simpul b.

3) Simpul terencil (*isolated vertex*)

Simpul terencil adalah simpul yang terhubung kepada simpul lain. Apabila direpresentasikan oleh sebuah gambar, simpul terencil adalah sebuah titik yang tidak memiliki sisi yang bersisian dengannya.

4) Graf kosong (*Empty graph*)

Graf kosong adalah graf yang tidak memiliki sisi ataupun simpul (himpunan kosong).

5) Derajat (*Degree*)

Derajat atau *degree* adalah jumlah sisi yang terhubung pada sebuah simpul. Pada gambar graf pertama pada gambar 2.2, simpul 1 memiliki nilai derajat dua. Sebab, simpul 1 bersisian dengan sisi (1, 2) dan sisi (1, 3).

6) Lintasan (*Path*)

Lintasan adalah sebuah urutan simpul dan sisi yang menggambarkan sebuah lintasan dari simpul awal (v_0) hingga simpul tujuan (v_n). Sebuah lintasan dari v_0 menuju v_n dapat ditulis seperti berikut:

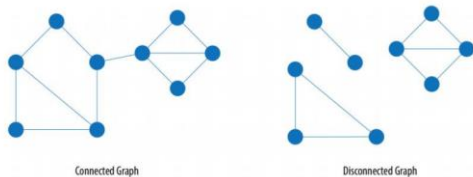
$$v_0, e_1, v_1, e_2, \dots, e_n, v_n$$

7) Siklus (*cycle*) atau sirkuit (*circuit*)

Sebuah lintasan yang dimulai dari sebuah simpul sembarang A dan berakhir pada simpul A juga disebut sebagai sebuah sirkuit atau siklus. Panjang sebuah sirkuit adalah jumlah sisi yang terdapat pada lintasan tersebut.

8) Keterhubungan (*connected*)

Dua buah simpul yang terhubung secara langsung ataupun tidak langsung (terdapat lintasan yang menghubungkan kedua simpul) menandakan bahwa kedua simpul tersebut memiliki sifat keterhubungan atau dua simpul tersebut dapat dinyatakan *connected*. Apabila sebuah simpul tidak terhubung secara langsung maupun tidak langsung dengan sebuah simpul lain, maka kedua simpul tersebut tidak keterhubungan (*disconnected*).



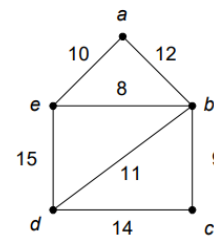
Gambar 2.6 Gambar graf keterhubungan dan graf tak-keterhubungan.

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

9) Graf berbobot (*weighted graph*)

Pada sebuah graf berbobot, setiap sisi memiliki sebuah nilai yang merepresentasikan hubungan berbobot antara dua buah simpul. Pembobotan sisi bermanfaat apabila dibutuhkan analisis lebih dalam terkait hubungan antara objek-objek diskrit. Contohnya, pada Gambar 2.1, setiap sisi dapat diberi sebuah bobot untuk merepresentasikan jarak antara kota-kota.



Gambar 2.7 Graf berbobot.

Sumber:

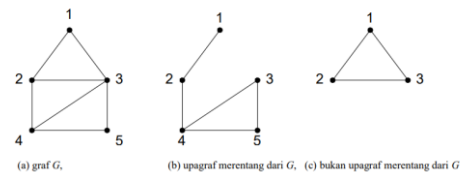
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

10) Upagraf (*Subgraph*)

Apabila terdapat sebuah graf G dengan $G = (V, E)$, dan sebuah graf $G_1 = (V_1, E_1)$ dengan $V_1 \subseteq V$ dan $E_1 \subseteq E$ maka G_1 merupakan upagraf dari G .

11) Upagraf merentang (*Spanning upagraph*)

Sebuah upagraf dikatakan upagraf merentang apabila semua simpul pada graf G terdapat pada upagraf G_1 .



Gambar 2.8 Upagraf merentang.

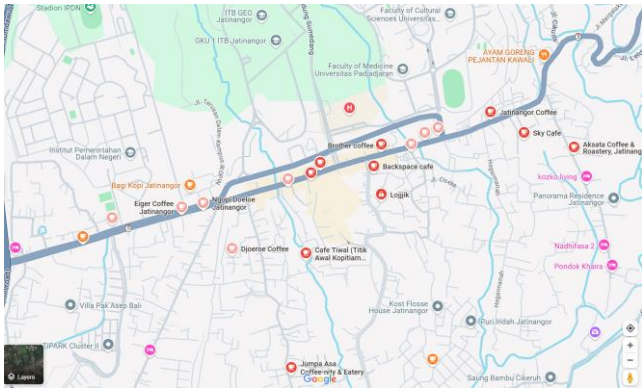
Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf>

III. PERCOBAAN

A. Data

Untuk percobaan, penulis memanfaatkan aplikasi google maps untuk mendapatkan informasi kafe-kafe di Jatinarong, termasuk nama kafe, rating kafe, total review, dan jarak kafe. Percobaan tidak menggunakan posisi langsung dari setiap kafe (tidak memanfaatkan latitude dan longitude) agar algoritma lebih sederhana. Fokus algoritma bukan pada ekstraksi dan pengolahan posisi terhadap lokasi, melainkan pada cara kerja sistem rekomendasi. Oleh karena itu, digunakan titik awal apartemen Pinewood Jatinarong untuk percobaan dan setiap atribut jarak pada data kafe dihitung berdasarkan titik awal tersebut (titik longitude dan latitude apartemen Pinewood: -6.935569169981525, 107.77176570020202).



Gambar 3.1 Wilayah yang digunakan untuk percobaan.
 Sumber: <https://maps.google.com/>

Data lokasi kafe seperti nama lokasi, rating, jarak, dan jumlah review didapatkan berdasarkan data/informasi terbaru yang disediakan pada google maps per tanggal 4 Januari 2025. Semua data kafe disimpan dalam sebuah file .json dengan format:

```
{ "nama": N, "rating": R, "jarak": J, "review count": RC }
```

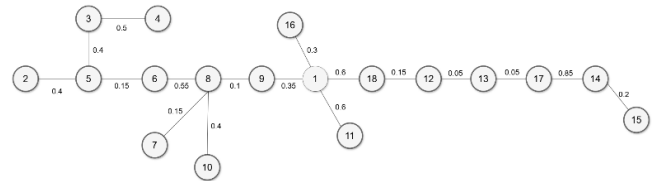
Satuan pada jarak adalah kilometer (km).

B. Pembobotan

Lokasi titik awal dengan lokasi setiap kafe diilustrasikan menggunakan graf berbobot. Setiap lokasi direpresentasikan dengan angka. Graf yang dibuat merupakan upagraf dengan bobot minimum untuk setiap sisinya (lintasan yang dipakai dari titik awal adalah jarak terdekat).

No	Kafe/Lokasi
1	Titik awal (acuan jarak)
2	Toleransi Kopi
3	Nomar Kopi
4	Bagi Kopi Jatininggor
5	Eiger Kopi Jatininggor
6	Ngopi Doeloe Jatininggor
7	Djoeroe Coffee
8	MinMax Coffee Jatininggor
9	Fore Coffee Jatininggor
10	Cafe Tiwal
11	Backspace Cafe
12	Eightfully Coffee & Milk Bar
13	TOMORO COFFEE – JATINANGOR
14	Jatininggor Coffee
15	Sky Cafe
16	Corner Coffee
17	Warung HiHu
18	Brother Coffee

Tabel 3.1 Lokasi dengan angka yang merepresentasikannya.
 Sumber: Dokumentasi pribadi



Gambar 3.2 Graf rekomendasi kafe Jatininggor dengan jarak sebagai bobot.

Sumber: Dokumentasi pribadi.

Pembobotan untuk sistem rekomendasi didapatkan menggunakan rumus berikut:

$$\text{Bobot} = \alpha \cdot d + \beta \cdot (1 - \text{AR})$$

Variabel α adalah faktor bobot untuk jarak, variabel d adalah jarak, variabel β adalah faktor bobot untuk adjusted rating, dan AR adalah Adjusted Rating. Adjusted rating (AR) dihitung dengan rumuse berikut:

$$\text{Adjusted Rating} = \left(\frac{\text{Rating} \cdot \text{Review Count}}{\text{Review Count} + K} \right)$$

Gambar 3.3 Rumus adjusted rating.

Sumber: Dokumentasi pribadi.

Rumus adjusted rating digunakan untuk mempertimbangkan jumlah review dalam penentuan bobot rating. Nilai K digunakan sebagai konstanta penyeimbang dalam perhitungan bobot rating. Konstanta K sendiri bernilai 70.

Nilai adjusted rating dinormalisasikan agar semakin besar nilai adjusted rating, semakin kecil nilai yang didapatkan dari bobot rating. Sebab, sistem rekomendasi merupakan implementasi mencari rekomendasi kafe yang diurut dari bobot terkecil.

Pembobotan yang dipakai untuk jarak adalah 0.5 dan adjusted rating juga 0.5. Mempertimbangkan daerah Jatininggor tidak luas dan jarak satu kafe ke kafe lain (jarak antar 2 simpul) rata-rata dekat. Rumus perhitungan bobot adalah sebagai berikut:

$$\text{Bobot} = 0.5 \cdot \text{Jarak} + 0.5 \cdot (1 - \text{Adjusted Rating})$$

Gambar 3.4 Rumus perhitungan bobot pada percobaan.

C. Algoritma

Percobaan sistem rekomendasi dibuat dalam program recSys.py menggunakan bahasa python.

```

1 import json
2
3 # Kelas untuk merepresentasikan data Cafe
4 class Cafe:
5     def __init__(self, nama, bobot):
6         self.nama = nama
7         self.bobot = bobot
8
9     def __repr__(self):
10        return f"({self.nama} (Bobot: {self.bobot:.4f}))"
11
12 # Fungsi untuk menghitung rating yang disesuaikan
13 def getAdjustedRating(rating, review_count, k=70):
14     return (rating + review_count) / (review_count + k)
15
16 # Fungsi untuk menghitung bobot berdasarkan formula
17 def getBobot(jarak, rating, review_count, k=70):
18     adjusted_rating = getAdjustedRating(rating, review_count, k)
19     return 0.5 * jarak + 0.5 * (1 - adjusted_rating / 5)
20
21 # Program utama
22 def main():
23     listKafe = [] # Daftar rekomendasi kafe
24
25     # Membaca file JSON
26     with open('cafe.json', 'r') as file:
27         data_kafe = json.load(file)
28
29     # Memproses setiap kafe
30     for kafe in data_kafe:
31         nama = kafe['nama']
32         jarak = kafe['jarak']
33         rating = kafe['rating']
34         review_count = kafe.get('review_count', 0) # Tambahkan jumlah review (default: 0)
35         bobot = getBobot(jarak, rating, review_count) # Hitung bobot
36         listKafe.append(Cafe(nama, bobot)) # Tambahkan ke daftar
37
38     # Mengurutkan daftar berdasarkan bobot terendah
39     listKafe.sort(key=lambda x: x.bobot)
40
41     # Menampilkan Hasil
42     i = 1
43     print("Hasil Rekomendasi:")
44     for kafe in listKafe:
45         print(f"{i}. {kafe}")
46         i += 1
47
48 # Menjalankan program
49 if __name__ == "__main__":
50     main()

```

Gambar 3.5 Program sistem rekomendasi kafe di Jatinangor.
 Sumber: Dokumentasi pribadi.

D. Hasil

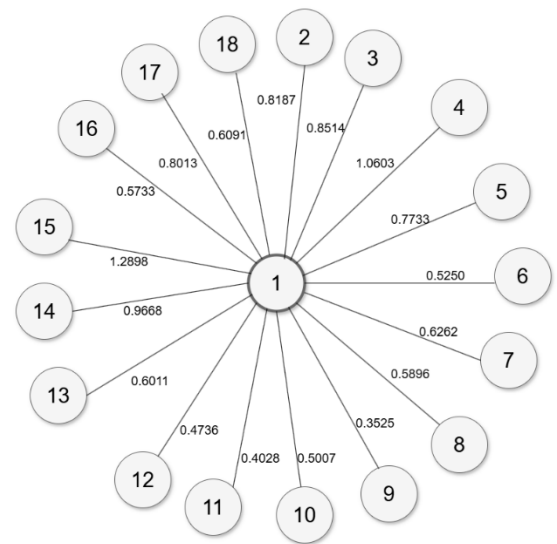
Berikut adalah output dari program sistem rekomendasi kafe:

```

Hasil Rekomendasi:
1. Fore Coffee (Bobot: 0.3525)
2. Backspace Cafe (Bobot: 0.4028)
3. Eightfully Coffee & Milk Bar (Bobot: 0.4736)
4. Cafe Tiwal (Bobot: 0.5007)
5. Ngopi Doeloe Jatinangor (Bobot: 0.5250)
6. Corner Coffee (Bobot: 0.5733)
7. MinMax Coffee Jatinangor (Bobot: 0.5896)
8. TOMORO COFFEE - JATINANGOR (Bobot: 0.6011)
9. Brother coffee (Bobot: 0.6091)
10. Djoeroe Coffee (Bobot: 0.6262)
11. Eiger Coffee Jatinangor (Bobot: 0.7733)
12. Warung HiHu (Bobot: 0.8013)
13. Toleransi Kopi (Bobot: 0.8187)
14. Nomar Kopi Roastery Jatinangor (Bobot: 0.8514)
15. Jatinangor Coffee (Bobot: 0.9668)
16. Bagi Kopi Jatinangor (Bobot: 1.0603)
17. Sky Cafe (Bobot: 1.2898)

```

Gambar 3.6 Output program recSys.py.
 Sumber: Dokumentasi pribadi.



Gambar 3.7 Graf berbobot dari output recSys.py (menggunakan representasi lokasi dengan angka dari tabel 3.1)

Sumber: Dokumentasi pribadi

IV. PEMBAHASAN

Program recSys.py berhasil memberikan output berdasarkan algoritma rekomendasi sistem yang telah dibuat. Output dari program menampilkan list hasil rekomendasi yang terurut dari bobot terkecil.

Pada graf berbobot (dengan bobot = jarak) pada gambar 3.2, Corner Coffee merupakan cafe terdekat dari titik awal dan Fore Coffee sebagai kafe terdekat kedua. Pada graf berbobot tersebut, Corner Coffee terurut sebagai cafe pertama. Namun, hasil program recSys.py menunjukkan Corner Coffee memiliki kedudukan keenam di bawah kafe Eightfully Coffee dalam list rekomendasi, walaupun Corner Coffee memiliki rating lebih bagus dan jarak lebih dekat daripada kafe Ngopi Doeloe Jatinangor.

```

1 {
2     "nama": "Corner Coffee",
3     "rating": 4.9,
4     "jarak": 0.3,
5     "review_count": 13
6 },

```

Gambar 3.8 Data kafe Corner Coffee.
 Sumber: Dokumentasi pribadi.

```

1 {
2     "nama": "Ngopi Doeloe Jatinangor",
3     "rating": 4.2,
4     "jarak": 0.85,
5     "review_count": 1402
6 },

```

Gambar 3.9 Data kafe Ngopi Doeloe Jatinangor.
 Sumber Dokumentasi pribadi.

Tetapi, kafe Ngopi Doeloe Jatinangor memiliki jumlah review count sebesar 1402, sedangkan Corner Coffee hanya memiliki review count total 13. Hal ini disebabkan oleh algoritma rekomendasi sistem yang mempertimbangkan jumlah review untuk pertimbangan bobot. Di sisi lain, Bagi Kopi Jatinangor mendapatkan posisi ke-16 pada list rekomendasi, walaupun rating yang dimiliki senilai 4.9 dengan total review count 272.

V. KESIMPULAN

Sistem rekomendasi kafe Jatinangor berdasarkan jarak dan rating kafe berhasil diimplementasikan pada program recSys.py. Sistem rekomendasi kafe dapat digunakan oleh mahasiswa Jatinangor atau penghuni Jatinangor lainnya yang ingin mencari kafe dengan pertimbangan jarak, rating, dan jumlah review yang dimiliki kafe. Namun, agar program dapat digunakan dengan lokasi titik awal yang dinamis, program perlu dimodifikasi dan ditambahkan fitur untuk menerima latitude dan longitude dan mengolah database yang dapat menyesuaikan atribut jarak dengan lokasi titik awal.

VI. UCAPAN TERIMA KASIH

Pertama, puji syukur dipanjatkan kepada Tuhan Yang Maha Esa karena atas izinnya, penulis dapat menyelesaikan makalah ini dengan kondisi sebaik mungkin. Penulis juga berterima kasih kepada orang tua, saudara, dan teman-teman yang telah mendukung penulis dalam pembuatan makalah ini. Ucapan terima kasih juga penulis beri kepada Dr. Ir. Rinaldi Munir, Dr. Nur Ulfa Maulidevi, S.T, M.Sc, dan Bapak Arrival Dwi Sentosa, S.Kom., M.T., selaku dosen mata kuliah Matematika Diskrit yang telah memberikan banyak ilmu yang bermanfaat yang penulis gunakan pada makalah ini dan juga pada perkuliahan. Terakhir, penulis memohon maaf apabila terdapat salah kata dalam penulisan makalah ini, segala hal yang telah ditulis dalam makalah bertujuan untuk memberi ilmu yang bermanfaat bagi pembaca. Penulis berharap makalah ini dapat bermanfaat bagi banyak orang.

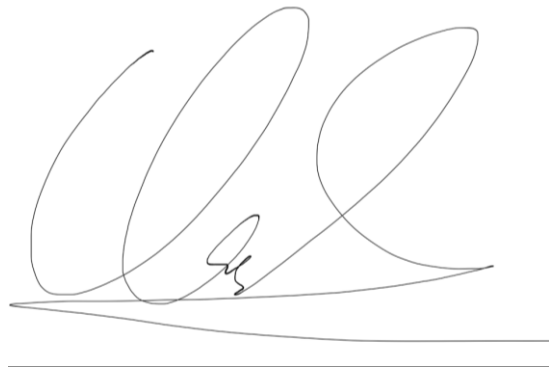
REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf> diakses pada 4 Januari 2025
- [2] <https://mti.binus.ac.id/2018/03/05/teori-graph-sejarah-dan-manfaatnya/> diakses pada 4 Januari 2025
- [3] [https://hive.telkomuniversity.ac.id/tribun-wiki-daftar-perguruan-tinggi-yang-berada-di-kawasan-pendidikan-jatinangor-tribun-jabar/#:~:text=Perguruan%20tinggi%20yang%20berada%20di%20kawasan%20Jatinangor%20bisa%20menjadi%20piliha,Manajemen%20Koperasi%20Indonesia%20\(Ikopin\)](https://hive.telkomuniversity.ac.id/tribun-wiki-daftar-perguruan-tinggi-yang-berada-di-kawasan-pendidikan-jatinangor-tribun-jabar/#:~:text=Perguruan%20tinggi%20yang%20berada%20di%20kawasan%20Jatinangor%20bisa%20menjadi%20piliha,Manajemen%20Koperasi%20Indonesia%20(Ikopin)) diakses pada 27 Desember 2025

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 Desember 2024



Mahesa Fadhillah Andre 13523140